Supporting Information

Quirks of Error Estimation in Cross-Linking/Mass Spectrometry

Lutz Fischer[1], *Juri Rappsilber*[1,2,*]

[1]Wellcome Trust Centre for Cell Biology, School of Biological Sciences, University of Edinburgh, Edinburgh EH9 3BF, United Kingdom.

[2]Department of Bioanalytics, Institute of Biotechnology, Technische Universität Berlin, 13355 Berlin, Germany.

* To whom correspondence should be addressed:

e-mail: Juri.Rappsilber@ed.ac.uk

False positive estimation for non-directional cross-links

For a non-directional cross-link, the cross-link of peptide A to peptide B is indistinguishable from a cross-link of peptide B to peptide A. This applies to the majority of currently used cross-linkers. Importantly, detection is based on mass spectrometric fragmentation spectra. Therefore, a cross-linker is non-directional if spectra of a cross-link A-B cannot be distinguished from those of B-A. Symmetric cross-linkers like BS^3 are always non-directional. Moreover even asymmetric cross-linkers like Succinimidyl-diazirine (SDA)[1] tend to be non-directional, as the cross-linker is rarely impacting on the fragmentation of peptides.

With T_{DB} being the size of the target database – T could be a peptide, linkage site or protein – we get for the target-target database (TT_{DB}):

1) $$TT_{DB} = \frac{{T_{DB}}^2 + T_{DB}}{2}$$

This corresponds to the green triangle in **Figure S1B**. Note that this is not a square as T_1 cross-linked to T_2 is identical to T_2 cross-linked to T_1, i.e. cross-links are not directional. The T_n, i.e. homo-dimerization of proteins is possible. For example, in a gedankenexperiment with two targets, A and B, TT_{DB} would have the size 3 (AA, AB, BB).

With D_{DB} being the size of the decoy database we get for the decoy-decoy database (DD_{DB}):

2) $$DD_{DB} = \frac{{D_{DB}}^2 + D_{DB}}{2}$$

This corresponds to the red triangle in **Figure S1B**, with identical considerations as for the target-target database.

Consequently, the size of the target-decoy database (TD_{DB}) is

3) $$TD_{DB} = T_{DB} \times D_{DB}$$

This corresponds to the orange square in **Figure S1B**. Every target needs to be paired with every decoy, as each is a unique pair.

Under the assumption that the target database has the same size as the decoy database we can replace T_{DB} and D_{DB} by A – representing the size of both the target and decoy database – and get:

4) $$TT_{DB} = \frac{A^2 + A}{2}$$

5) $$TD_{DB} = A^2$$

and

6) $$DD_{DB} = \frac{A^2 + A}{2}$$

To estimate the false positives (FP) among the target-target matches (TT) we need to estimate both the number of matches with both peptides incorrectly identified (false identification) among the target-target matches ($ff(TT)$) (red curve, **Figure S1D**) and the number with only one peptide incorrectly identified ($tf(TT)$) (orange curve, **Figure S1D**).

7) $FP(TT) = tf(TT) + ff(TT)$

As both the size of the TT and the DD database is the same and assuming the random match probability of TT and DD to be identical, a match with both peptides being wrong has the same chance to match both databases. As there is no other way to match DD than both partners being wrong (red curve, **Figure S1E**) we can assume:

8) $ff(TT) = ff(DD) = DD$

For matches with one correctly identified (true) peptide (tf) one peptide has to be from the target database. The chance of the second peptide to hit the target database is the same as hitting the decoy database. Therefore, the number of tf to TT should be the same as the number of tf matches to TD (orange curves, **Figure S1D, E**)

9) $tf(TT) = tf(TD)$

The sum of all TD matches is the sum of matches with one correctly identified (true) peptide ($tf(TD)$) and matches with both peptides incorrectly identified (false/random) ($ff(TD)$):

10) $TD = tf(TD) + ff(TD)$

And therefore $tf(TD)$ is

11) $tf(TD) = TD - ff(TD)$

Therefore, we need to estimate $ff(TD)$. We can also use the Decoy-Decoy matches here. The chance for a match with two false peptide identifications, to randomly fall in either TD_{DB} or DD_{DB}, is only dependent on the size of these databases. Therefore the $ff(TD)$ should be proportional to DD:

12) $ff(TD) = k \times DD$

with k being the proportionality factor.

Using this with formula 11 gives:

13) $tf(TD) = TD - ff(TD) = TD - k \times DD$

If we extend formula 7 with 9, 12 and 13 we get the false positives among the □□ matches as:

14) $FP(TT) = ff(TT) + tf(TT) = DD + (TD - k \times DD)$

This can be reduced to:

15) $FP(TT) = TD + DD(1 - k)$

As there is no difference between the target and decoy for a random match, the probability for $ff(TD)$ and $ff(DD)$ should be defined by the sizes of the underlying databases. Therefore, the proportionality factor k can be determined as:

16) $TD_{DB} = k \times DD_{DB}$

Adding here formulas 5 and 6 lead to

17) $A^2 = k \times \frac{A^2 + A}{2}$

And therefore:

18) $k = \frac{2 \times A^2}{A^2 + A}$

Using (formula 5) this resolves to

19) $k = \frac{2 \times TD_{DB}}{TD_{DB} + \sqrt{TD_{DB}}}$

So we get for the estimated false positives in TT:

20) $FP(TT) = TD + DD\ (1 - 2\frac{TD_{DB}}{TD_{DB} + \sqrt{TD_{DB}}})$

False positive estimation for directional cross-links

For a directional cross-linker, the fragmentation spectra of cross-link A-B differ significantly from those of B-A. One instance in which this could be the case is if a structurally asymmetric MS2-cleavable cross-linker were to be employed[2]. Linker-containing fragments would differ in mass, depending on which end of the cross-linker was attached. Such a cross-linker might also impact on the success by which fragments of one or the other peptide are observed. Consequently, the chance of identifying A-B would be different from that of identifying B-A. The same then applies to random matches. The random space for two wrongly identified peptides would therefore be quadratic (**Figure S1C**). The search space divides into the following areas:

21) $TT_{DB} = A^2$

22) $TD_{DB} = 2 * A^2$

23) $DD_{DB} = A^2$

Joining that with formula 16 results in

24) $2 \times A^2 = k_{directional} \times A^2$

Therefore, we get $k_{directional} = 2$. If we join that with formula 15 we get

25) $FP_{directional}(TT) = TD + DD(1 - 2)$

and

26) $FP_{directional}(TT) = TD - DD$

Impact of using the wrong formula

The main difference between the two formulas is the size of the factor *k*. While for the directional case k is always 2, for the non-directional case k is dependent on the size of the database. If the wrong formula is used, the maximal error one makes is therefore directly dependent on the size of the database (**Figure S2**). For the currently more common case of a non-directional cross-linker, the use of a directional cross-linker very rapidly approaches the correct estimate with increasing database size. Already with 200 entries (e.g. 200 unique linkable residues in target and decoy) the error is smaller than 1%. Moreover, the error applies only to the scaling of decoy-decoy-matches. Therefore, it only comes into play once the first decoy-decoy match is observed. For PSMs, peptide pairs, and residue pairs this is not likely to be of practical significance. For protein pairs, however, it will affect the results when working with protein complexes for example.

xiFDR – a software that boosts CLMS results and applies FDR cut-offs

The software, xiFDR, is a search-tool-independent application to calculate FDR for CLMS. The application also maximizes the number of returned hits for a desired FDR. xiFDR applies a stepwise FDR: it first filters the PSMs to a specified FDR and then aggregate the PSMs to unique combinations of peptide pairs, filters these again by an FDR and aggregate the peptide pairs that pass the FDR to unique residue pairs (=cross-links). The peptide-pair score is calculated by the following formula:

27) $$S_{PP} = \sqrt{\sum({S_{PSM}}^2)}$$

Where S_{PSM} is the score of a supporting PSM. By using the square root of the sum of squares we emphasize higher scoring matches. We sum the top-scoring PSM for each combination of peptide pair and charge state, as PSMs of different charge states add information while PSMs of the same charge state essentially replicate a search result, be it correct or false. The same formula is then used stepwise to aggregate the score of lower-level matches to higher-level matches. So we can generalize formula 27 to:

28) $$S_{higher\ level} = \sqrt{\sum({S_{lower\ level}}^2)}$$

xiFDR then applies the specified peptide-pair FDR and aggregates the results into unique residue pairs. These get scored in the same way as the peptide pairs and filtered to the specified link-FDR. The same process is then repeated for protein pairs.

Matches can be divided into different groups with inherently different chances of being right or wrong. As was previously observed[3], PSMs and peptide pairs have a different a priori probability to be correct – depending on whether they are linking a protein with itself (self-link) or between proteins (between link). Additionally, peptide-length groups can optionally be defined. In our workflow, smaller peptides tend to have a higher chance of being wrong than longer peptides, as was observed for protein identification at least in the case of Mascot and Andromeda[4]. Since we are looking at cross-linked peptides, and the shorter peptide is often the less well defined one, the peptide-length grouping is performed on the shorter peptide. For residue-pair level and protein-pair level only the distinction of *'self-link'* versus *'between'* is used.

As the pre-filtering on lower levels can have a big impact on the number of results passing the higher-level FDR the application provides a means for an automated optimization of lower-level FDRs to maximize the results that pass a defined higher-level FDR. To do so, xiFDR searches through combinations of lower-level FDRs with a defined number of steps for each FDR level. It then takes the FDR combinations that result in the highest number of results and searches around these values for an improvement. The optimization stops and reports the "best" settings once the number of hits can no longer be improved. The optimization strategy employed is not perfect and can get stuck in local minima. Even so, the number of identifications at a given confidence can often be increased (for example, **Figure 3B**).

xiFDR accepts csv-files as input. The csv-file must contain a set of columns (see **Table S1**). The user interface is shown in **Figure S4**. The first step is to load a csv-file. If the column names deviate from those predefined in the software the user is prompted to select the appropriate columns. This should make it possible to use xiFDR with any cross-link database search software that can produce a csv-output of matches or for which exists a converter to generate csv-files. Furthermore, xiFDR supports the upcoming mzIdentML[5] standard 1.2. It expects cross-links to be marked as was discussed at the HUPO PSI meeting April 2013. To identify two peptides, that were identified as part of a cross-link, the according SpectrumIdentificationItem(s) have to have a cvParam with accession number MS:1002511 and the ID of that cross-link as value. Additionally, the cross-linker has to be defined on both peptides as a modification with two cvParam, MS:1002509 on one peptide and MS:1002510 the second peptide, denoting the linkage sites on both peptides. For reading and writing mzIdentML we use the jmzidentml[6] library in version 1.1.3.

Once the file is loaded the user can define any combination of FDRs for each level and filter their input-data to the specified FDRs. If the only FDRs of interest concern residue pairs or protein pairs then the application offers to "maximize" residue pairs or protein pairs as described previously. For the grouping of PSM-matches and peptide pairs the length steps can be defined.

Once the FDRs are calculated the result tab provides a summary of all levels. A "+" button beside each FDR-level can be pressed to provide a more detailed report of each sub-group in the

input. As we group matches of one level and do a separate FDR calculation on each group, the final reported FDR and range is defined as a weighted average of the sub-FDRs and ranges. Weighting is done based on how many matches passed the FDR within each group.

To save results, a base-folder has to be selected and a name for the result-files. The program provides a set of csv-files, with one file for each level of information: a list of all unique protein pairs that passed the successive FDR-cut-offs; a file for all unique links that constitute these protein pairs; a file for the unique peptide pairs; a file for all passed PSM and a file that reports all unique proteins. Additionally, a summary-file is generated providing an overview of how many entries passed what level and how much they were filtered down by higher level-FDRs – split up by considered groups. If the input file was an mzIdentML-file also an mzIdentML-file can be generated as output.

Resolution of FDR estimation

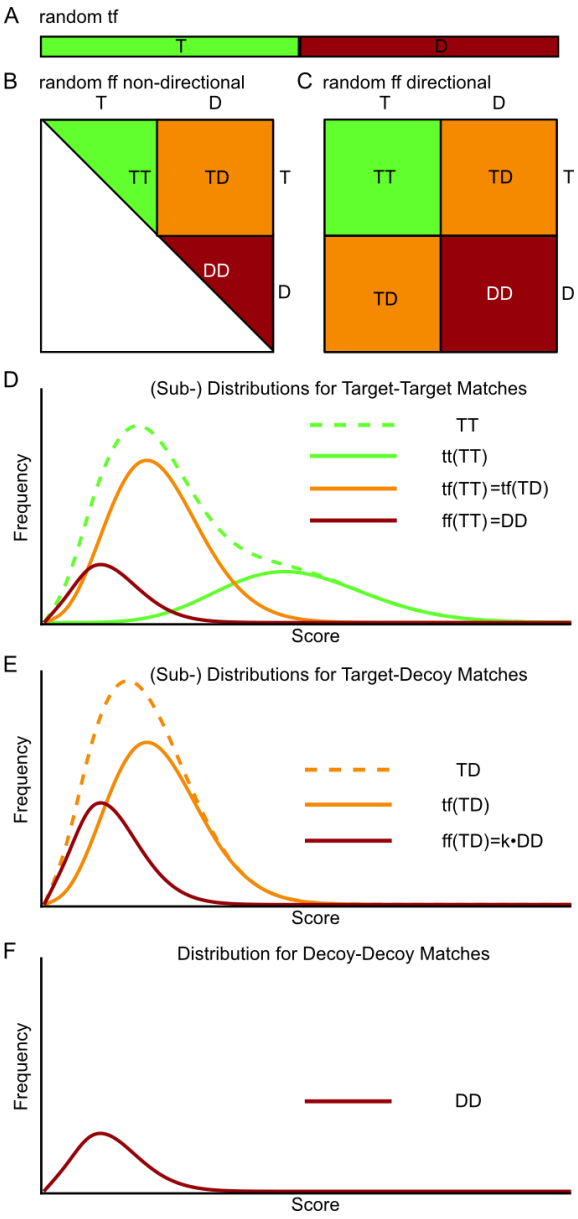
When filtering for an FDR the accuracy of the estimate is mainly defined by how many decoy and target matches one sees. Especially for low FDR-values accuracy is impaired by sparse amount of decoy matches (**Figure 4**). Until now this was mostly ignored. We propose to report the size of the window around the target FDR. This window is defined by the next higher FDR that can be calculated (occurrence of the next decoy) minus the next lower FDR. E.g. 1%[0.72%] instead of just 1%. In doing so we can provide at least some information as to what the actual measurable FDRs are. While this is not the same as an actual accuracy it gives at least an

indication of an expected true FDR. For the aforementioned results of 5% FDR this would mean 5%[1.4%] for the optimized case and 5%[1.3%] the for the non-pre-filtered case.

Column	Required	Description	
Run	o	Some description of the LCMS acquisition	Either psmid or Run and Scan have to be present.
Scan	o	scan-number for the MS2-event	
psmid	o	a unique id referencing the peptide spectrum match	
peptide1	X	the sequence of peptide1	
peptide2	x	the sequence of peptide2	
peptide length 1		the length of peptide 1	If the length is not provided the string-length of sequence is used
peptide length 2		the length of peptide 2	
peptide link 1	x	at which residue in the peptide 1 is the linker attached	
peptide link 2	x	at which residue in the peptide 2 is the linker attached	
is decoy 1	x	is the first peptide coming from a decoy-protein	
is decoy 2	x	is the second peptide coming from a decoy protein	
precursor charge	x	what was the precursor charge state of the PSM	
score		A single score for the whole peptide	either a single score defined for the whole spectrum match or a separate score for each peptide is required
score ratio		how to split the score to define the "support" of each matched protein	
Peptide1 Score		Score of peptide 1	
Peptide2 Score		Score of peptide 2	
accession1	x	accessions numbers for the protein that peptide 1 could belong to	If a peptide can be found in several proteins (or several times in one protein), then for each time it appears in a protein that protein accession number has to be named as many times as the peptide appears. The peptide positions have to matches to the protein accession numbers
accession2	x	accessions numbers for the protein that peptide 2 could belong to	
description1		A description for the first protein	
description2		A description for the second protein	
peptide position 1	x	where in the first protein is the peptide located	
peptide position 2	x	where in the second protein is the peptide located	

Table S1: Descriptions of columns for the CSV-import of peptide spectrum matches

Figure S1



Random Spaces and Distributions

A) Random search space for cross-linked peptides (peptide pairs) with one true and one false peptide identification. *B*) Random search space for peptide pairs with both peptides wrongly identified for non-directional cross-linkers. *C*) Same as B but for directional cross-linkers. *D*) Schematic histogram of target-target matches (dashed line) and the constituting sub-distributions: tt(TT) matches (green line) with two correctly identified peptides. tf(TT) matches (orange line) with only one correctly identified peptide. ff(TT) matches (red line) with two incorrectly identified cross-links. *E*) Schematic histogram of target-decoy matches (dashed line) and the constituting sub-distributions: tf(TD) matches (orange line) with only one correctly identified peptide. ff(TD) matches (red line) with two incorrectly identified cross-links. As one peptide is a decoy every target-decoy match has at least 1 wrongly identified peptide. *F*) Schematic histogram of decoy-decoy matches. As both peptides are decoy peptides there is no other sub-distribution. Panels D, E, and F are valid for both directional and non-directional cross-linkers. Only the relative size of ff(TD) would be different between these two cases. T, Target; D, Decoy; t, true; f, false

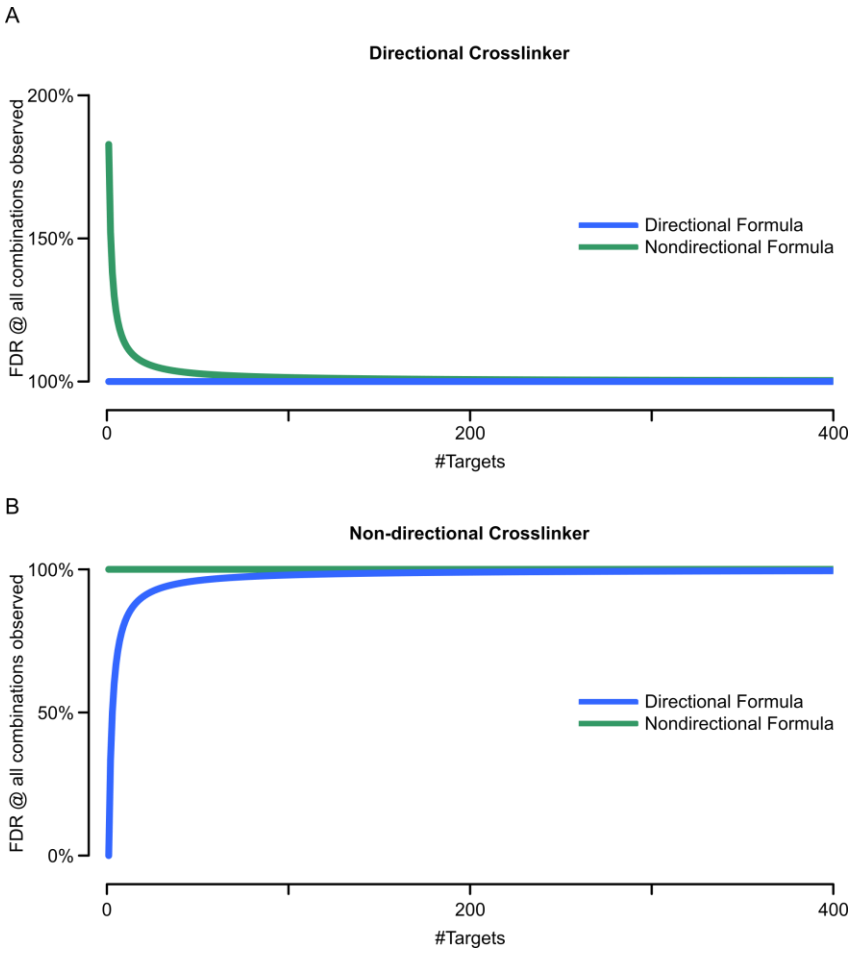


Figure S2: Error when using the different formulas for FDR estimation

The estimated FDR (y-axis) assuming all possible combinations in a database of a given size (x-axis) e.g. all possible peptide pairs, residue-pairs or protein-pairs are found ones. The FDR in that case should be 100%. The first value is a database of size 2. The results of both formulas are shown – green if the non-directional formula is used and blue if the directional formula is used. A) Estimated FDR, if a directional cross-linker is used (e.g. MS2-cleavable cross-linker with an asymmetric cleavage site) B) Estimated FDR for a non-directional cross-linker (e.g. BS^3).

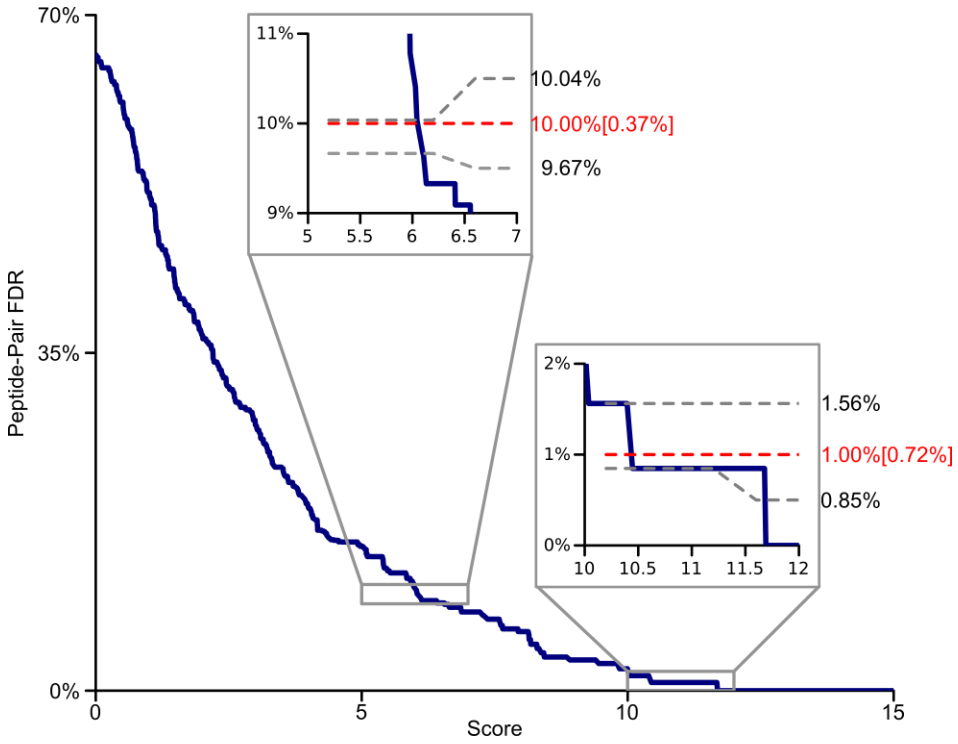


Figure S3: Resolution of FDR Estimation

The chart shows the residue-pair FDR over score. The inlets show a zoom in around two FDR-values. The lower inlet shows a 1% target FDR and the actual measurable FDRs around 1% of 0.85% and 1.56%. The upper inlet shows a 10% target FDR and the measurable window. For 10% the window is a half the width of the window at 1%, indicating that we have a better resolution and consequentially would expect a higher accuracy for stating a 10% FDR then for stating a 1% FDR.

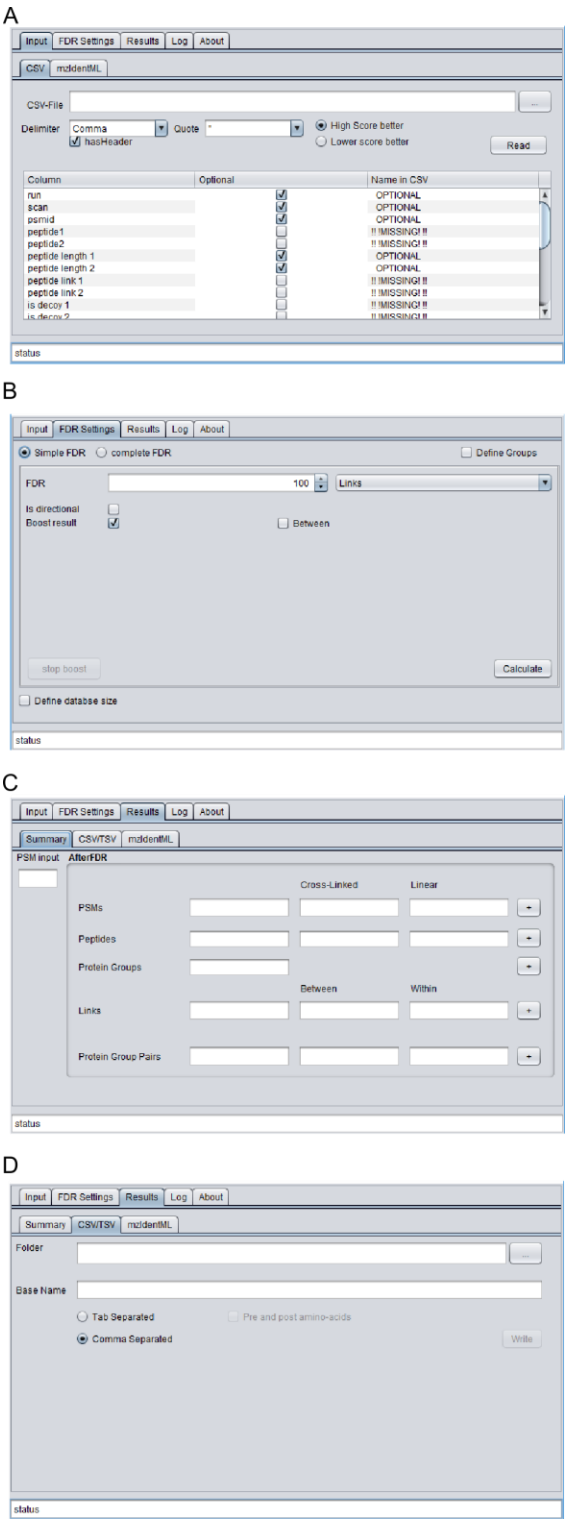


Figure S4: Interface of xiFDR

A) Input tab of xiFDR for reading CSV-files (CSV=character separated values). Columns found in the file are matched to the expected columns (lower part), which is done automatically by xiFDR if the naming convention (**Table 1**) is observed or manually if not. *B*) Submission tab to define the target FDR. A secondary tab with access to all settings can be accessed from here (not shown). *C*) Summary tab provides an overview of those identifications passing the FDR-filter. More detailed information about each level can be displayed with the [+]-buttons. *D*) Interface for writing out the results as a CSV-file. The folder can be selected and a base-name that is appended to each file of the output.

SUPPLEMENTAL REFERENCES

(1) Belsom, A.; Schneider, M.; Fischer, L.; Brock, O.; Rappsilber, J. *Mol. Cell. Proteomics* **2016**, *15*, 1105-1116.

(2) Liu, F.; Goshe, M. B. *Anal. Chem.* **2010**, *82*, 6215-6223.

(3) Walzthoeni, T.; Claassen, M.; Leitner, A.; Herzog, F.; Bohn, S.; Forster, F.; Beck, M.; Aebersold, R. *Nat. Methods* **2012**, *9*, 901-903.

(4) Cox, J.; Mann, M. *Nat. Biotechnol.* **2008**, *26*, 1367-1372.

(5) Jones, A. R.; Eisenacher, M.; Mayer, G.; Kohlbacher, O.; Siepen, J.; Hubbard, S. J.; Selley, J. N.; Searle, B. C.; Shofstahl, J.; Seymour, S. L.; Julian, R.; Binz, P. A.; Deutsch, E. W.; Hermjakob, H.; Reisinger, F.; Griss, J.; Vizcaino, J. A.; Chambers, M.; Pizarro, A.; Creasy, D. *Mol. Cell. Proteomics* **2012**, *11*, M111 014381.

(6) Reisinger, F.; Krishna, R.; Ghali, F.; Rios, D.; Hermjakob, H.; Vizcaino, J. A.; Jones, A. R. *Proteomics* **2012**, *12*, 790-794.